Lecture 8:
Sequential Networks and Finite State Machines

CSE 140: Components and Design Techniques for Digital Systems

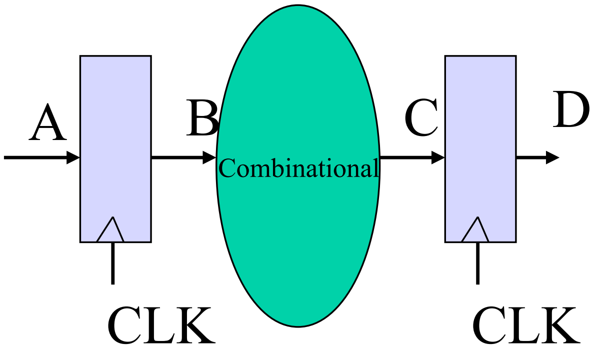
Spring 2014

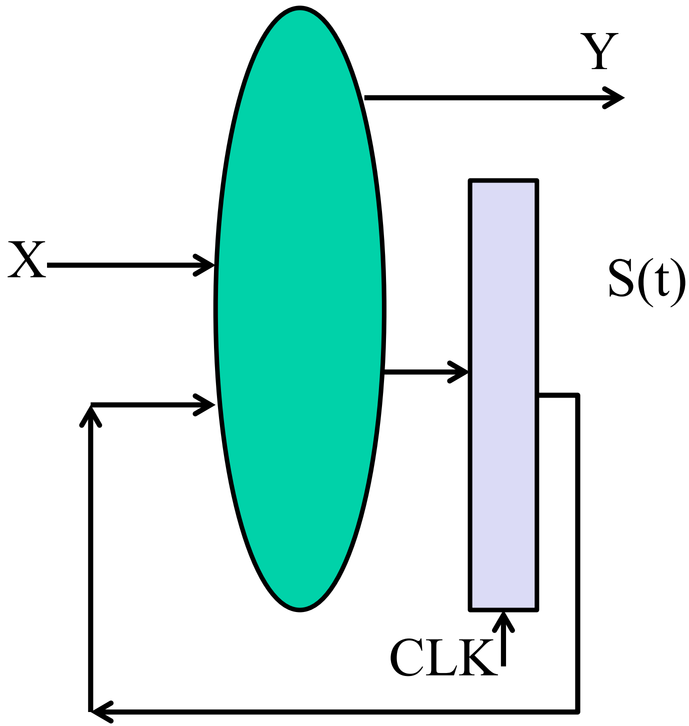
CK Cheng, Diba Mirza

Dept. of Computer Science and Engineering

University of California, San Diego

Sequential Networks





1. Components F-Fs
2. Specification
3. Implementation: Excitation Table

Specification

- Combinational Logic
  - Truth Table
  - Boolean Expression
  - Logic Diagram (No feedback loops)
- Sequential Networks: State Diagram (Memory)
  - State Table and Excitation Table
  - Characteristic Expression
  - Logic Diagram (FFs and feedback loops)

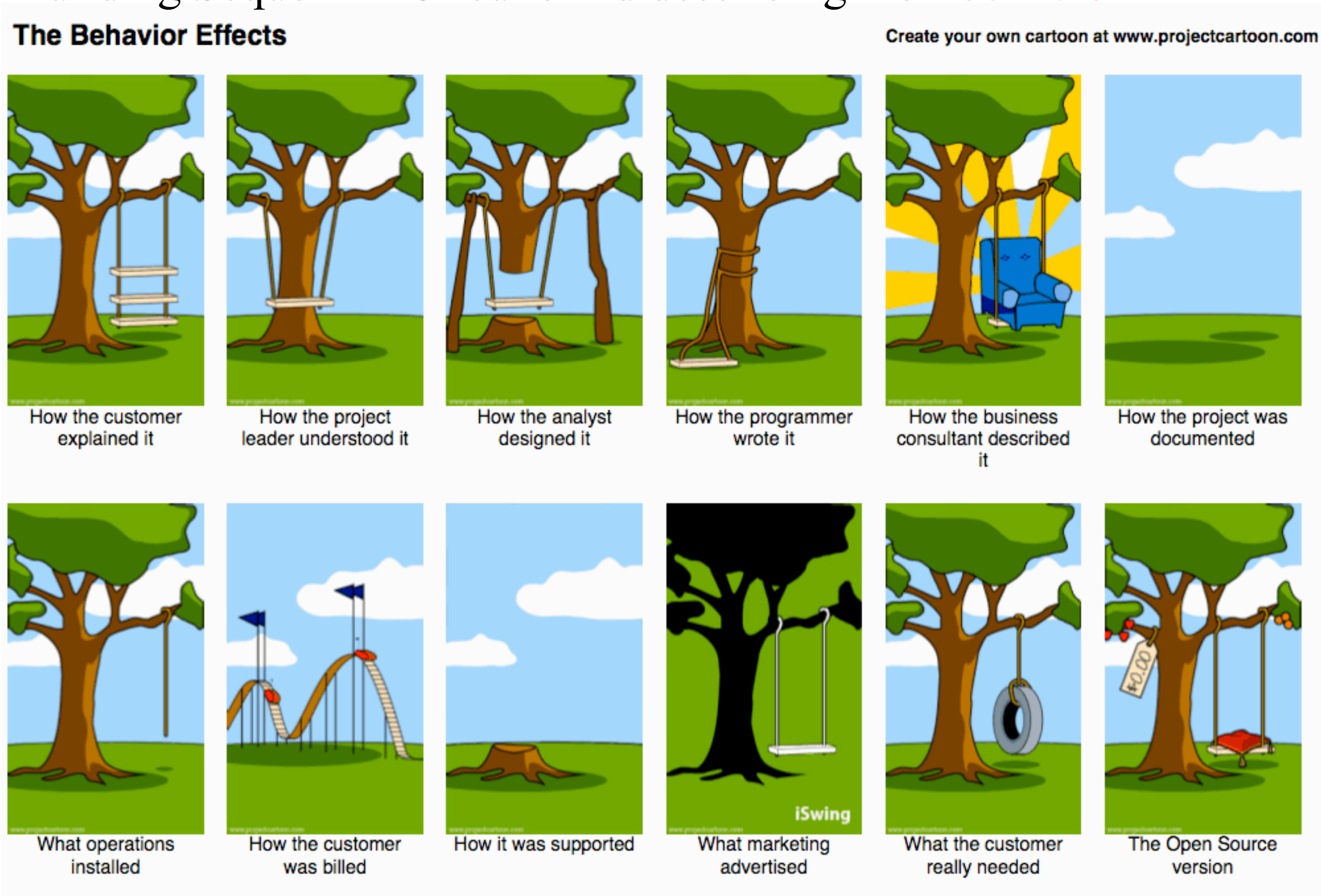
Specification: Finite State Machine

- Input Output Relation
- State Diagram (Transition of States)
- State Table
- Excitation Table (Truth table of FF inputs)
- Boolean Expression
- Logic Diagram

Specification: Examples

- Transition from circuit to finite state machine representation
  - Netlist => State Table => State Diagram => Input Output Relation
- Example 1: a circuit with D Flip Flops
- Example 2: a circuit with other Flip Flops

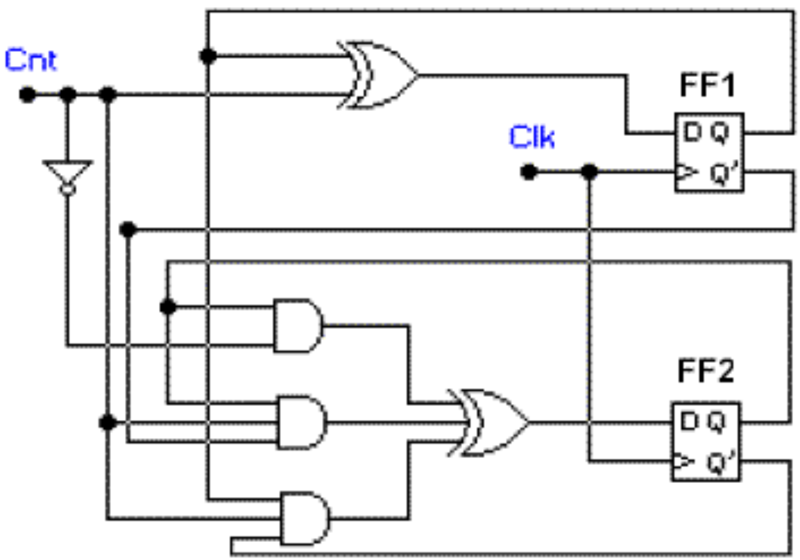
Building Sequential Circuits and describing their behavior



What we will learn:

1. Given a sequential circuit, describe its behavior over time
2. Given the behavior of a sequential circuit, implement the circuit

Sequential Circuit: Wall-E

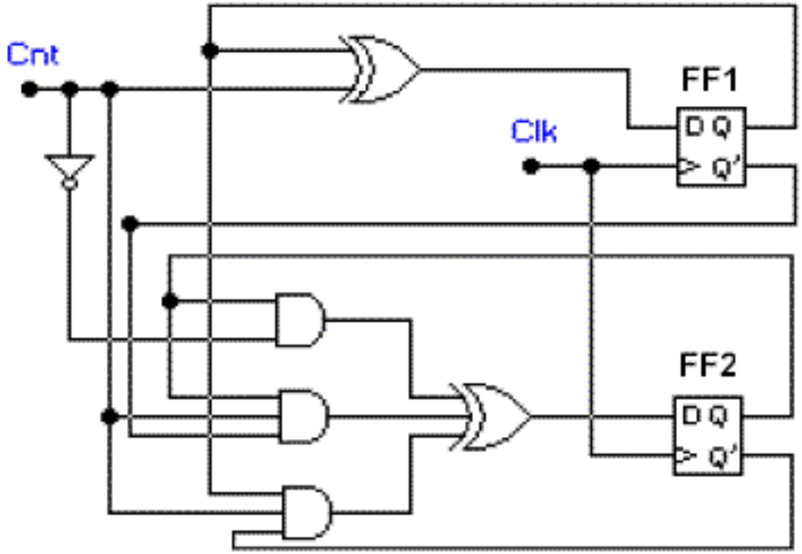


How does Wall-E behave?



What does it mean to describe the behavior of a sequential circuit

Specify how the output of the circuit changes as a function of inputs and the state of the circuit



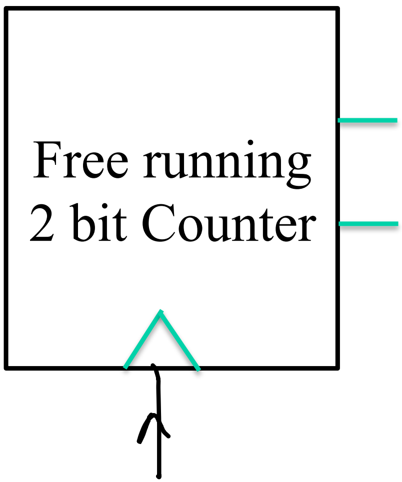


PI Q: What is the difference between the state of a circuit and its output?

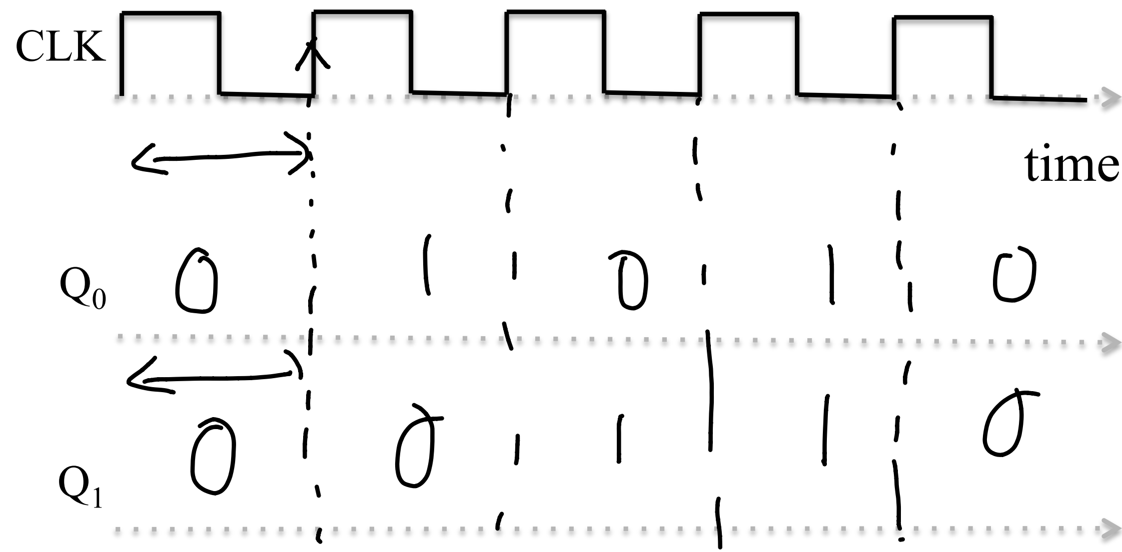
A. The output is independent of the state
B. The output and state are the same thing
C. The state is special type of output that is fed back into the circuit
D. The state is input information that is independent of previous outputs

State: What is it? Why do we need it?

Symbol/ Circuit



Behavior over time

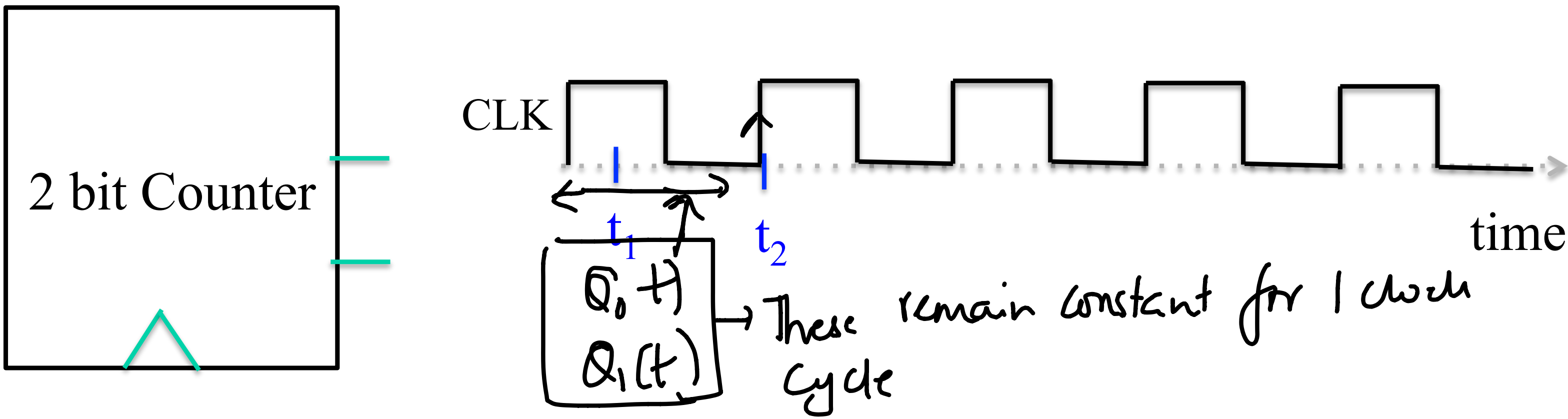


What is the expected output of the counter over time?

State: What is it ? Why do we need it?

Symbol/ Circuit

Behavior over time

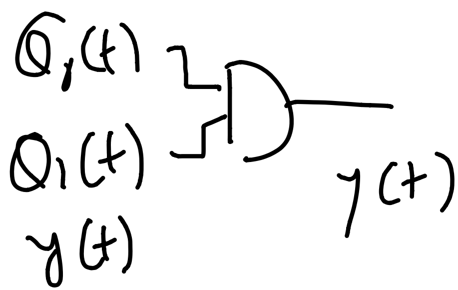


PI Q: At time t_1, what information is needed to produce the output of the counter at the next rising edge of the clock (i.e t_2)?

A. All the outputs of the counter until t_1
B. The initial output of the counter at time t=0
C. The output of the counter at current time t_1
D. We cannot determine the output of the counter at t_2 prior to t_2

State: What is it ? Why do we need it?

- The state is distilled output information that tells us everything we need to know to produce the next output. That is why it is fed back into the circuit.
- In the case of the 2-bit counter the output (i.e. the current count) is also the state of the counter. But we could have had other outputs that were not part of the state. E.g. A signal that indicated whether the current count is greater than 2.



Finite State Machines: Describing circuit behavior over time

Finite State Machine

Symbol/ Circuit

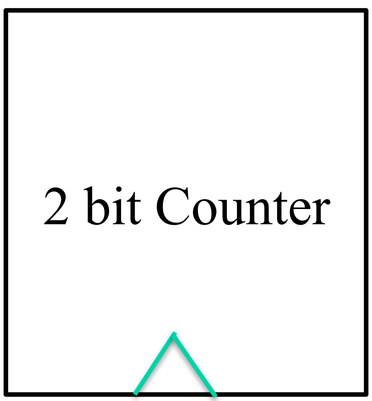
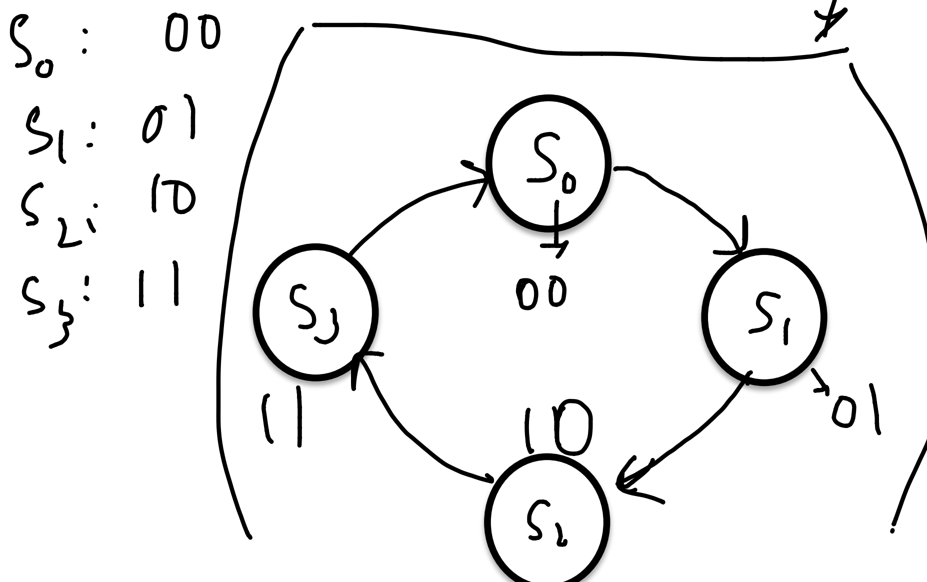


Diagram that depicts behavior over time

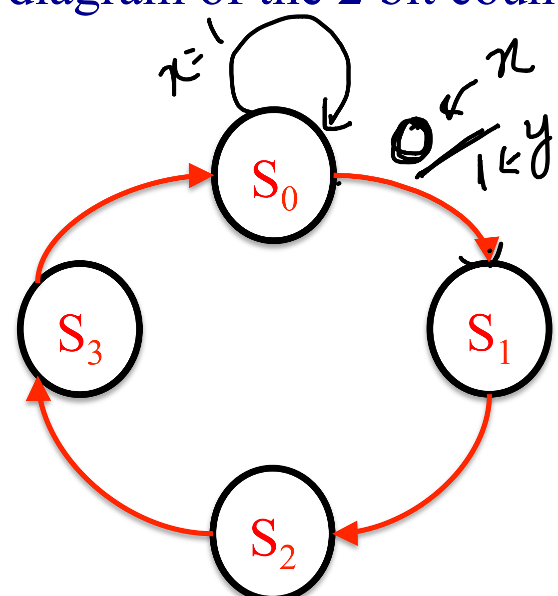


State Diagrams: Describing circuit behavior over time

PI Q: What information is not explicitly indicated in the state diagram?

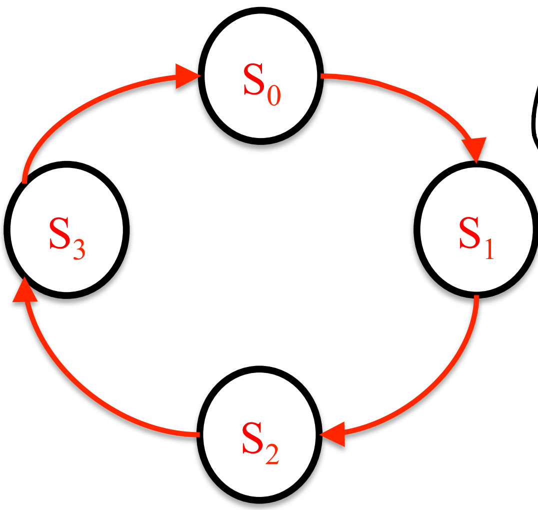
A. The input to the circuit
B. The output of the circuit
C. The time when state transitions occur
D. The current state of the circuit.
E. The next state of the circuit.

State diagram of the 2 bit counter



Finite State Machine

Implementing the 2 bit counter



State Diagram

State encoding $(Q_1(t), Q_0(t))$

$S_0 = (0,0)$
$S_1 = (0,1)$
$S_2 = (1,0)$
$S_3 = (1,1)$

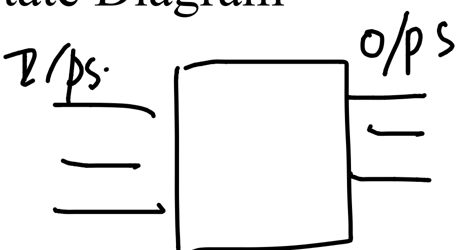
Inputs | Outputs

Current state	Next State
S_0	S_1
S_1	S_2
S_2	S_3
S_3	S_0

↓ Substitute S_0, S_1, S_2, S_3

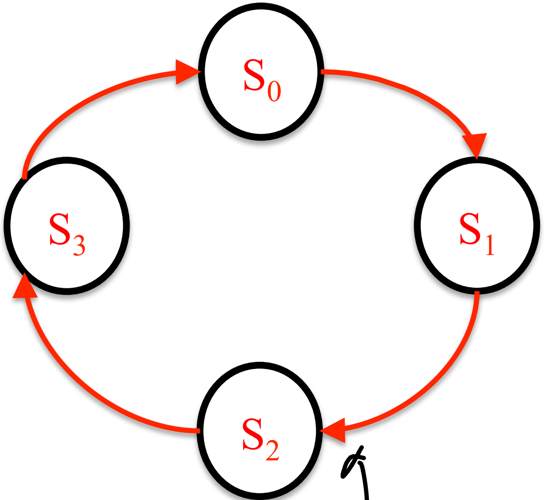
$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

State Table

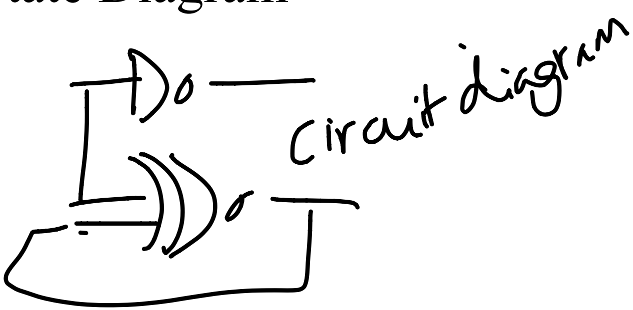


Write state diagram in i/p output form

Implementing the 2 bit counter



State Diagram



Current state	Next State
S_0	S_1
S_1	S_2
S_2	S_3
S_3	S_0

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

State Table

Implementing the 2 bit counter

Inputs | | Outputs Y(t) | Z(t)

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1 ←
0	1	1 ←	0
1	0	1 ←	1 ←
1	1	0	0

State Table

PI Q: To obtain the outputs $Q_0(t+1)$ and $Q_1(t+1)$ from the inputs $Q_1(t)$ and $Q_0(t)$ we need to use:

A. Combinational logic 60
B. Some other logic 40

$$Y(t) = Q_1(t) \otimes Q_0(t) \qquad \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0$$

$$Z(t) = \bar{Q}_0(t)$$

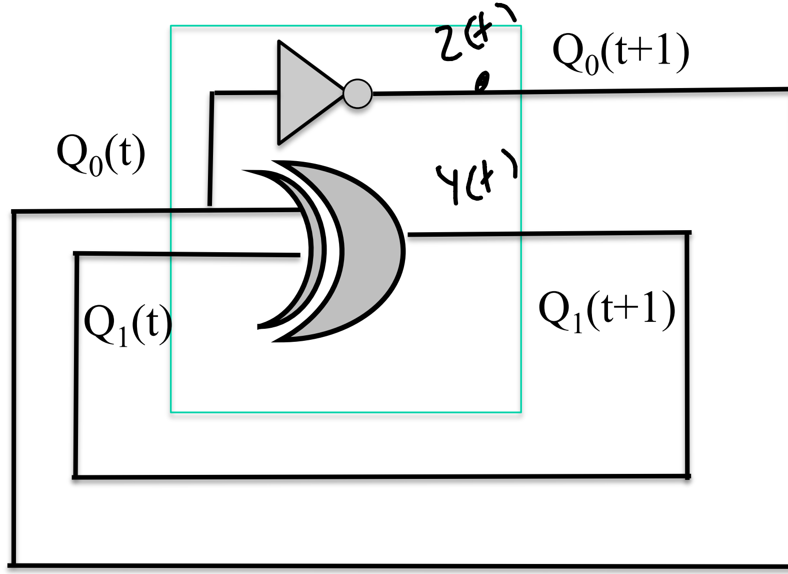
If we use D-flip flops, the next state follows (Y(t), Z(t))

$$Q_1(t+1) = Y(t) \qquad Q_0(t+1) = Z(t)$$

Implementing the 2 bit counter

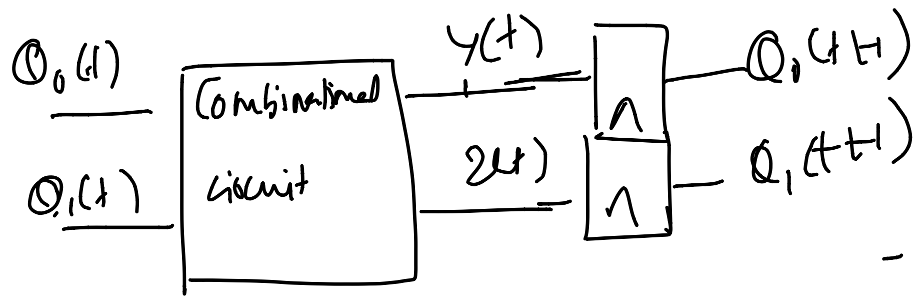
$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

State Table

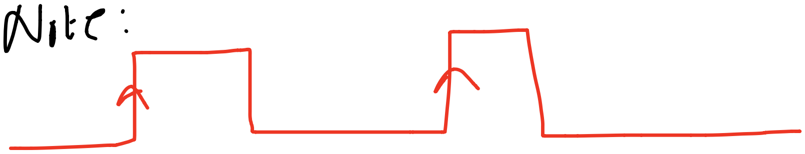


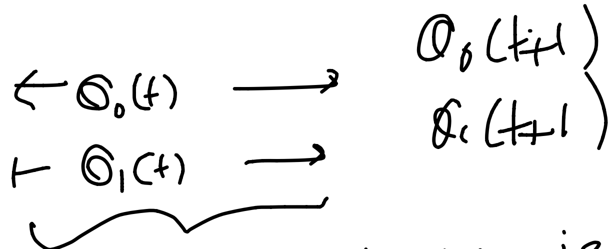
PI Q: What is wrong with the 2-bit counter implementation shown above

A. The combinational circuit is incorrect
B. The circuit state changes correctly but continuously rather than at the rising edge of the clock signal
C. The output of the circuit is unreliable because inputs can get corrupted



Note:





Based on the current state, i.e. $(Q_0(t), Q_1(t))$, in this clock cycle we will compute the inputs to the flip flops $(Y(t), Z(t))$ which will determine the value of the state in the next clock cycle i.e. $(Q_0(t+1), Q_1(t+1))$.

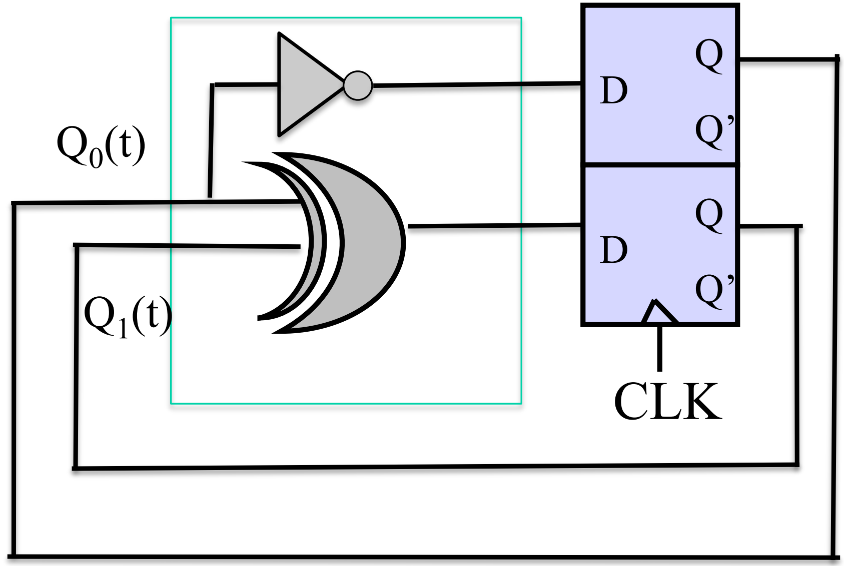
Implementing the 2 bit counter

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

State Table

$Q_0(t+1) = Q_0(t)'$

$Q_1(t+1) = Q_0(t)\,Q_1(t)' + Q_0(t)'\,Q_1(t)$



Implementation of 2-bit counter

We store the current state using D-flip flops so that:

- The inputs to the combinational circuit don't change while the next output is being computed
- The transition to the next state only occurs at the rising edge of the clock

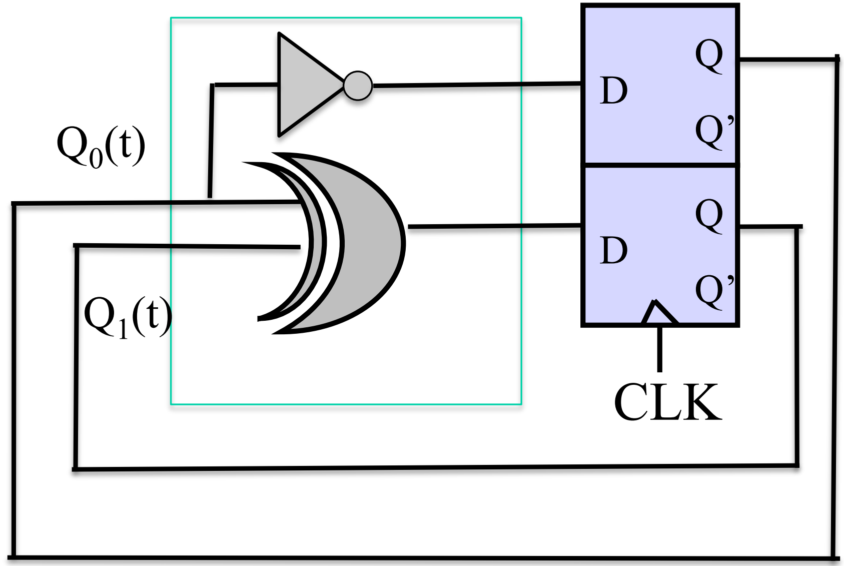
Implementing the 2 bit counter

$Q_1(t)$	$Q_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

State Table

$Q_0(t+1) = Q_0(t)'$

$Q_1(t+1) = Q_0(t)\, Q_1(t)' + Q_0(t)'\, Q_1(t)$

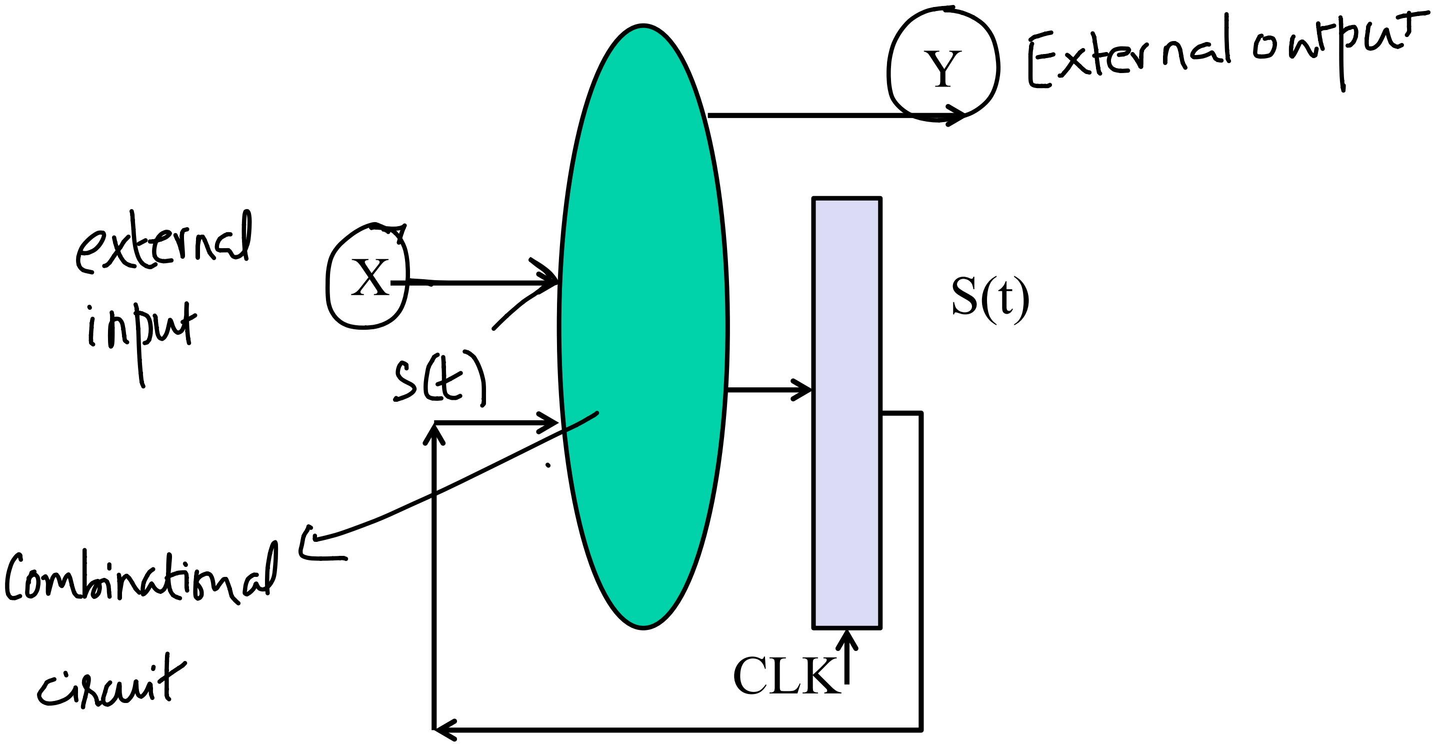


Implementation of 2-bit counter

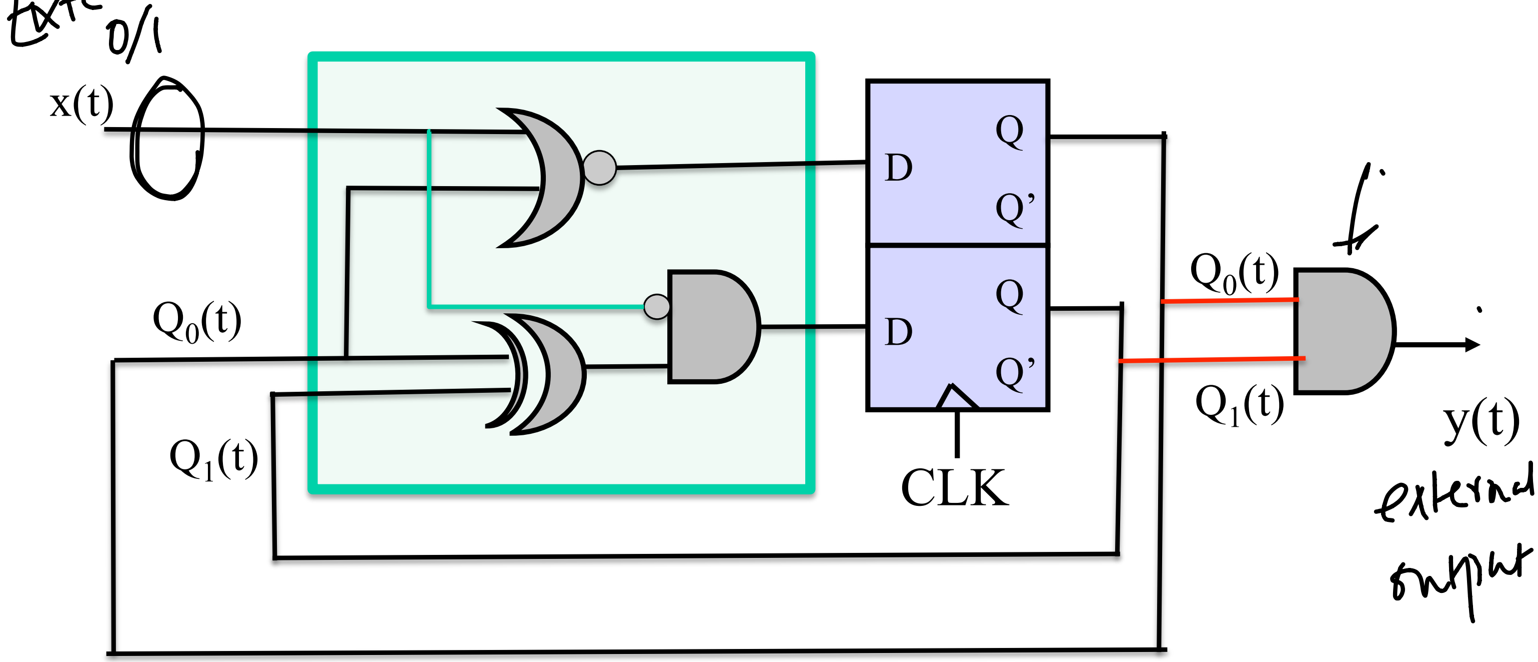
PI Q: When did we fix our choice of flip flops in the design process?

A. When we drew the state diagram
B. When we wrote down the state table
C. When wrote the characteristic expression
D. When we implemented the circuit from the characteristic expression

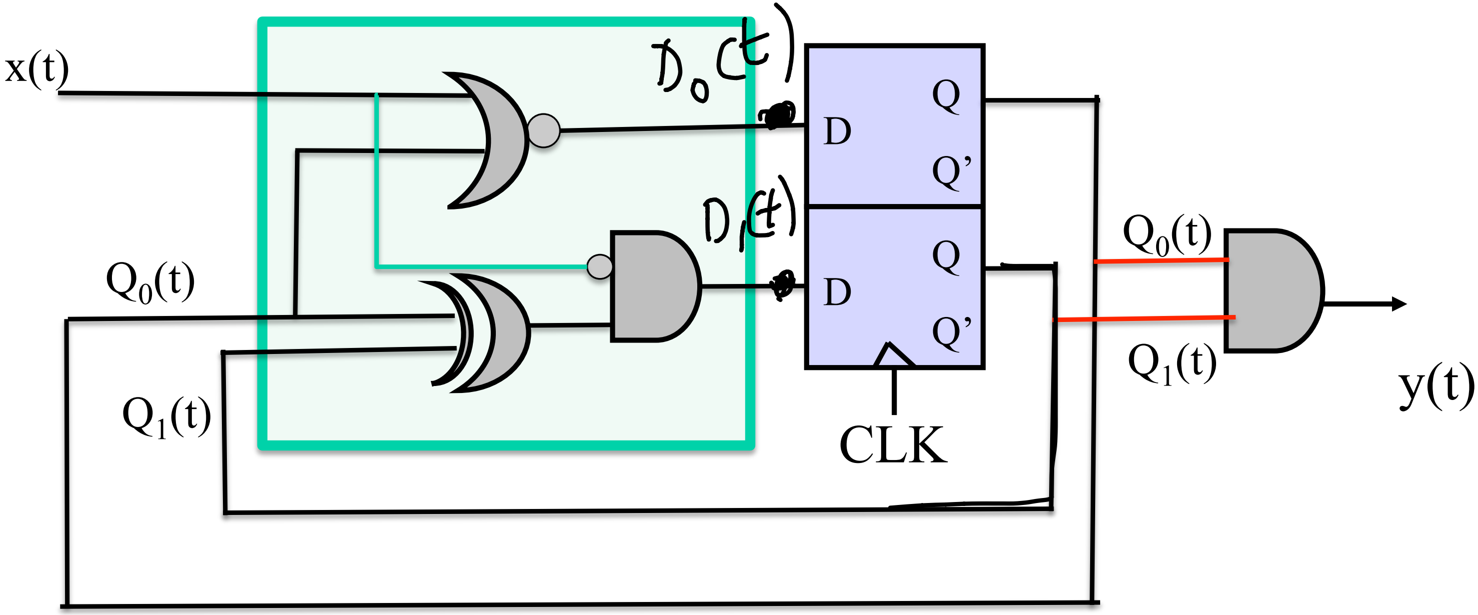
Generalized Model of Sequential Circuits



Modified 2 bit counter



Modified 2 bit counter



Characteristic Expression:

$y(t) = Q_0(t)\,Q_1(t)$

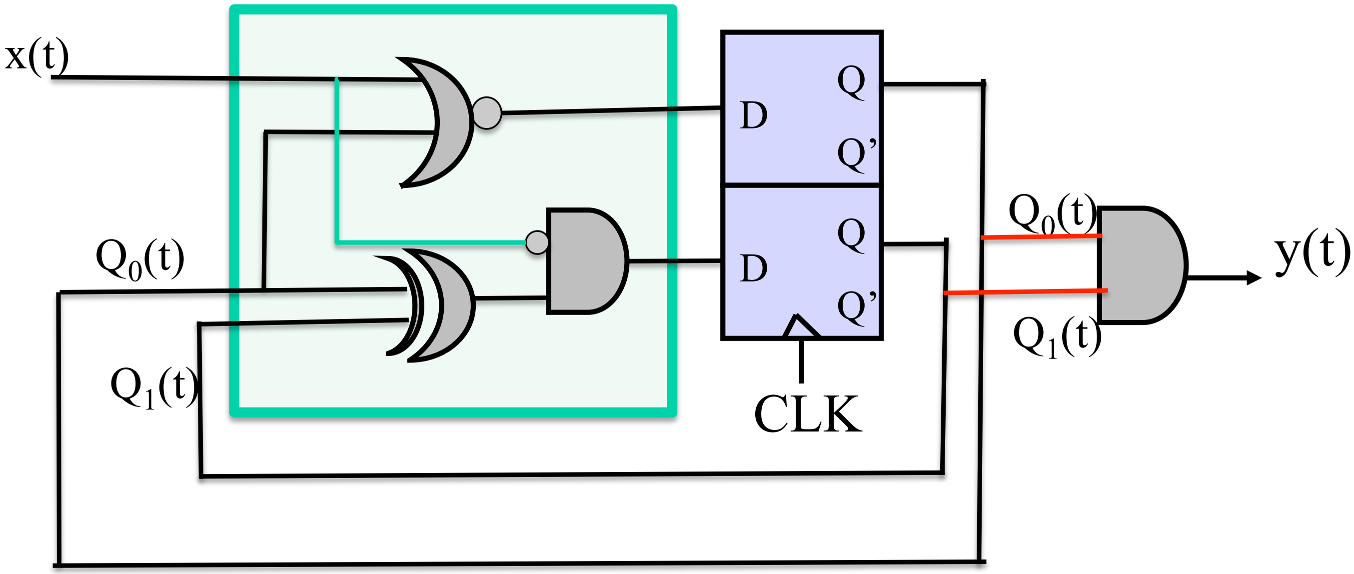
$Q_0(t+1) = D_0(t)$

$Q_1(t+1) = D_1(t)$

$D_0(t) = \overline{x(t)} \cdot \overline{Q_0(t)}$

$D_1(t) = \overline{x}\,[Q_0(t) \oplus Q_1(t)$

Modified 2 bit counter



$y(t) \ = \ Q_1(t)Q_0(t)$

$Q_0(t+1) = \ D_0(t) = x(t)' \, Q_0(t)'$

$Q_1(t+1) = \ D_1(t) = x(t)'(Q_0(t) \oplus Q_1(t))$

Netlist ⇔ State Table ⇔ State Diagram ⇔ Input Output Relation

Characteristic Expression:

$$y(t) = Q_1(t)Q_0(t)$$
$$Q_0(t+1) = D_0(t) = x(t)' \, Q_0(t)'$$
$$Q_1(t+1) = D_1(t) = x(t)'(Q_0(t) \oplus Q_1(t))$$

State table

PS \ input	x=0	x=1
→ 0 0	(0 1, 0)	(0 0, 0)
0 1	(1 0, 0)	(0 0, 0)
1 0	(1 1, 0)	(0 0, 0)
1 1	(0 0, 1)	(0 0, 1)

$Q_1(t)\ Q_0(t)$ | [$(Q_1(t+1)\ Q_0(t+1), y(t))$]

Present State | Next State, Output

State Assignment

S_0	00
S_1	01
S_2	10
S_3	11

PS \ input	x=0	x=1
S_0	$(S_1, 0)$	$(S_0, 0)$
S_1	$(S_2, 0)$	$(S_0, 0)$
S_2	$(S_3, 0)$	$(S_0, 0)$
S_3	$(S_0, 1)$	$(S_0, 1)$

Netlist ⇔ State Table ⇔ State Diagram ⇔ Input Output Relation

$$y(t) = Q_1(t)Q_0(t)$$
$$Q_0(t+1) = D_0(t) = x(t)' Q_0(t)'$$
$$Q_1(t+1) = D_1(t) = x(t)'(Q_0(t) \oplus Q_1(t))$$

State table

PS \ input	x=0	x=1
0 0	01, 0	00, 0
0 1	10, 0	00, 0
1 0	11, 0	00, 0
1 1	00, 1	00, 1

$Q_1(t)$ $Q_0(t)$ | $Q_1(t+1)$ $Q_0(t+1)$, $y(t)$
Present State | Next State, Output

State Assignment

Let:
$S_0 = 00$
$S_1 = 01$
$S_2 = 10$
$S_3 = 11$

⇒

PS \ input	x=0	x=1
S_0	S_1, 0	S_0, 0
S_1	S_2, 0	S_0, 0
S_2	S_3, 0	S_0, 0
S_3	S_0, 1	S_0, 1

Remake the state table using symbols instead of binary code , e.g. ’00’ .

Netlist ⇔ State Table ⇔ State Diagram ⇔ Input Output Relation

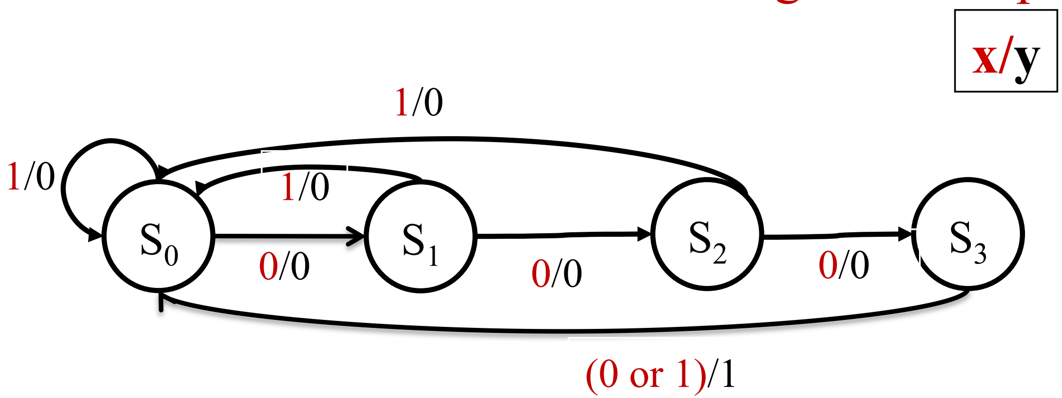


PS \ input	x=0	x=1
S_0	S_1, 0	S_0, 0
S_1	S_2, 0	S_0, 0
S_2	S_3, 0	S_0, 0
S_3	S_0, 1	S_0, 1

Given inputs and initial state, derive output sequence

Time	0	1	2	3	4	5
Input	0	1	0	0	0	-
State	S0					
Output						

Netlist ⇔ State Table ⇔ State Diagram ⇔ Input Output Relation

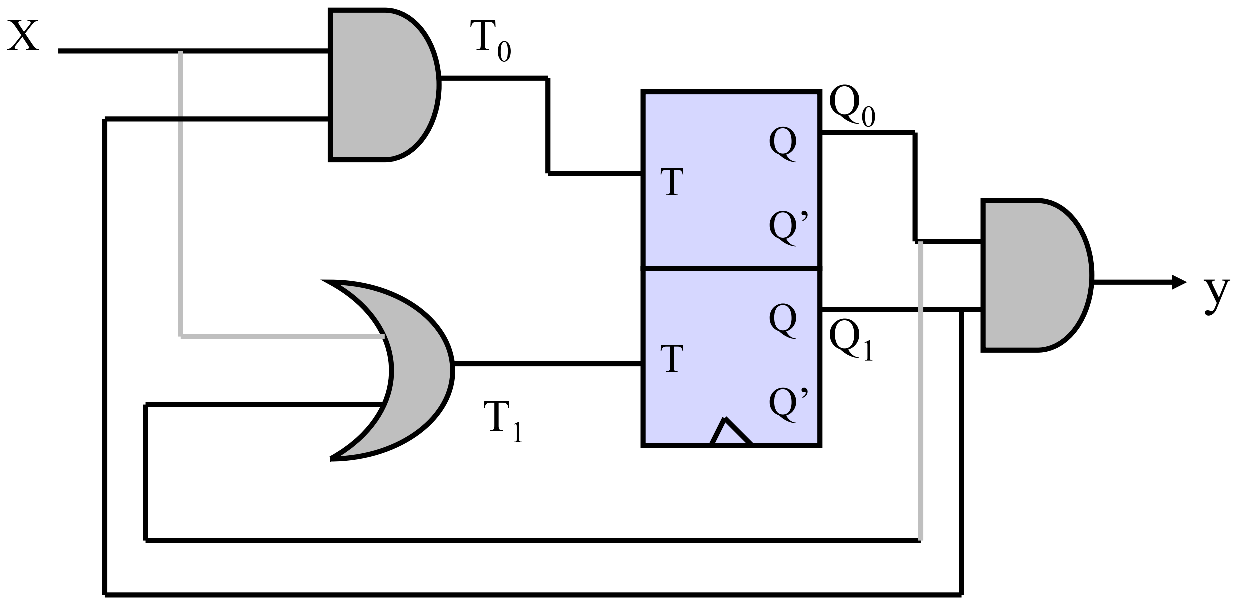


PS \ input	x=0	x=1
S_0	S_1, 0	S_0, 0
S_1	S_2, 0	S_0, 0
S_2	S_3, 0	S_0, 0
S_3	S_0, 1	S_0, 1

Example: Given inputs and initial state, derive output sequence

Time	0	1	2	3	4	5
Input	0	1	0	0	0	-
State	S0	S1	S0	S1	S2	S3
Output	0	0	0	0	0	1

Example 3 Circuit with T Flip-Flops



$$y(t) = Q_1(t)Q_0(t)$$
$$T_0(t) = x(t)\,Q_1(t)$$
$$T_1(t) = x(t) + Q_0(t)$$

Logic Diagram => Excitation Table => State Table

$y(t) = Q_1(t)Q_0(t)$
$T_0(t) = x(t)\ Q_1(t)$
$T_1(t) = x(t) + Q_0(t)$
$Q_0(t+1) = T_0(t)\ Q'_0(t)+T'_0(t)Q_0(t)$
$Q_1(t+1) = T_1(t)\ Q'_1(t)+T'_1(t)Q_1(t)$

Excitation Table:
Truth table of the F-F inputs

id	$Q_1(t)$	$Q_0(t)$	x	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0
2	0	1	0	1	0	1	1	0
3	0	1	1	1	0	1	1	0
4	1	0	0	0	0	1	0	0
5	1	0	1	1	1	0	1	0
6	1	1	0	1	0	0	1	1
7	1	1	1	1	1	0	0	1

Excitation Table: iClicker

In excitation table, the inputs of the flip flops are used to produce

A. The present state
B. The next state

Excitation Table =>State Table => State Diagram

id	$Q_1(t)$	$Q_0(t)$	x	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0
2	0	1	0	1	0	1	1	0
3	0	1	1	1	0	1	1	0
4	1	0	0	0	0	1	0	0
5	1	0	1	1	1	0	1	0
6	1	1	0	1	0	0	1	1
7	1	1	1	1	1	0	0	1

State Assignment
S0 00
S1 01
S2 10
S3 11

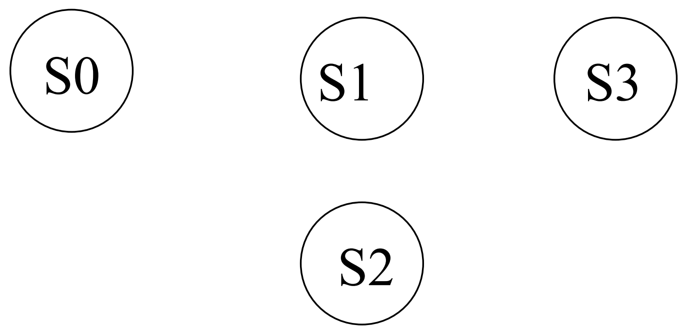
PS\Input	X=0	X=1
S0		
S1		
S2		
S3		

Excitation Table =>State Table => State Diagram

id	$Q_1(t)$	$Q_0(t)$	x	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0
2	0	1	0	1	0	1	1	0
3	0	1	1	1	0	1	1	0
4	1	0	0	0	0	1	0	0
5	1	0	1	1	1	0	1	0
6	1	1	0	1	0	0	1	1
7	1	1	1	1	1	0	0	1

State Assignment
S0 00
S1 01
S2 10
S3 11

PS\Input	X=0	X=1
S0	S0,0	S2,0
S1	S3,0	S3,0
S2	S2,0	S1,0
S3	S1,1	S0,1

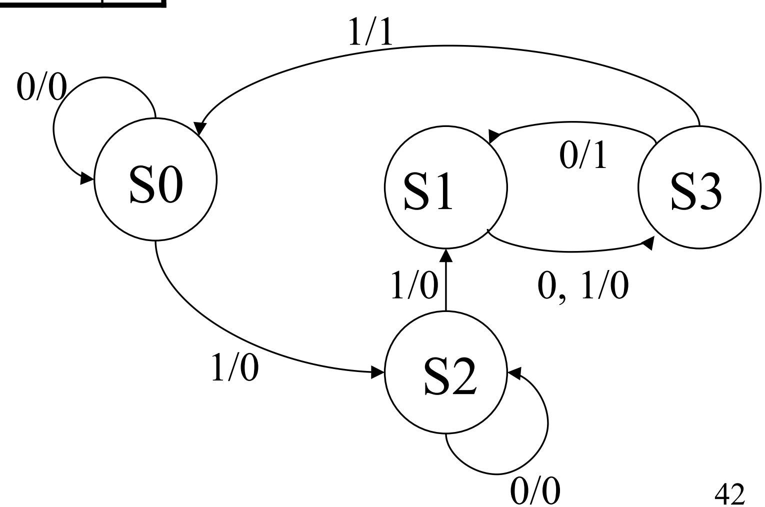


Excitation Table =>State Table => State Diagram

id	$Q_1(t)$	$Q_0(t)$	x	$T_1(t)$	$T_0(t)$	$Q_1(t+1)$	$Q_0(t+1)$	y
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	1	0	0
2	0	1	0	1	0	1	1	0
3	0	1	1	1	0	1	1	0
4	1	0	0	0	0	1	0	0
5	1	0	1	1	1	0	1	0
6	1	1	0	1	0	0	1	1
7	1	1	1	1	1	0	0	1

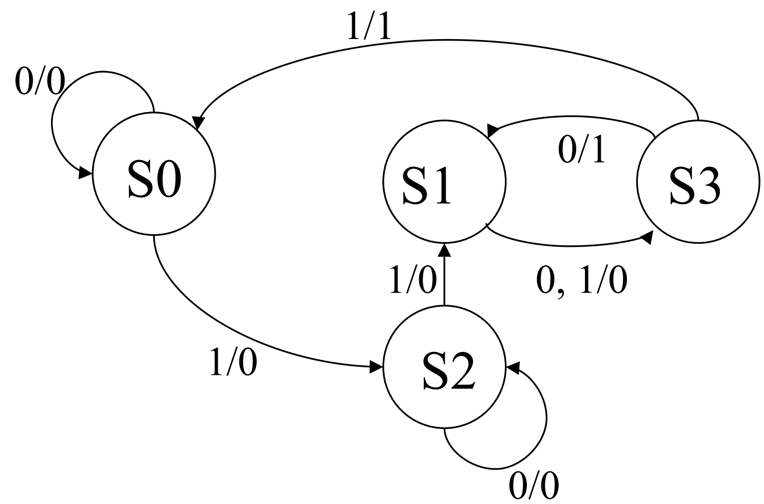
State Assignment
S0 00
S1 01
S2 10
S3 11

PS\Input	X=0	X=1
S0	S0,0	S2,0
S1	S3,0	S3,0
S2	S2,0	S1,0
S3	S1,1	S0,1



Netlist ⇔ State Table ⇔ State Diagram ⇔ Input Output Relation

PS\Input	X=0	X=1
S0	S0,0	S2,0
S1	S3,0	S3,0
S2	S2,0	S1,0
S3	S1,0	S0,1

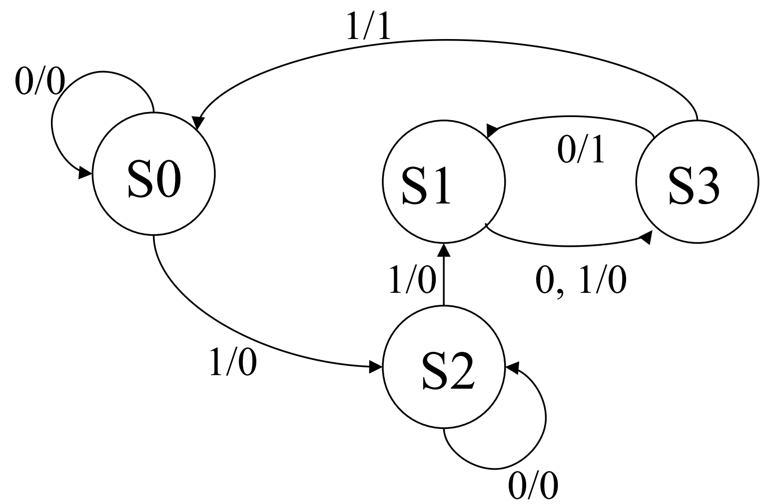


Example: Output sequence

Time	0	1	2	3	4	5
Input	0	1	1	0	1	-
State	S0					
Output						

Netlist ⇔ State Table ⇔ State Diagram ⇔ Input Output Relation

PS\Input	X=0	X=1
S0	S0,0	S2,0
S1	S3,0	S3,0
S2	S2,0	S1,0
S3	S1,0	S0,1



Example: Output sequence

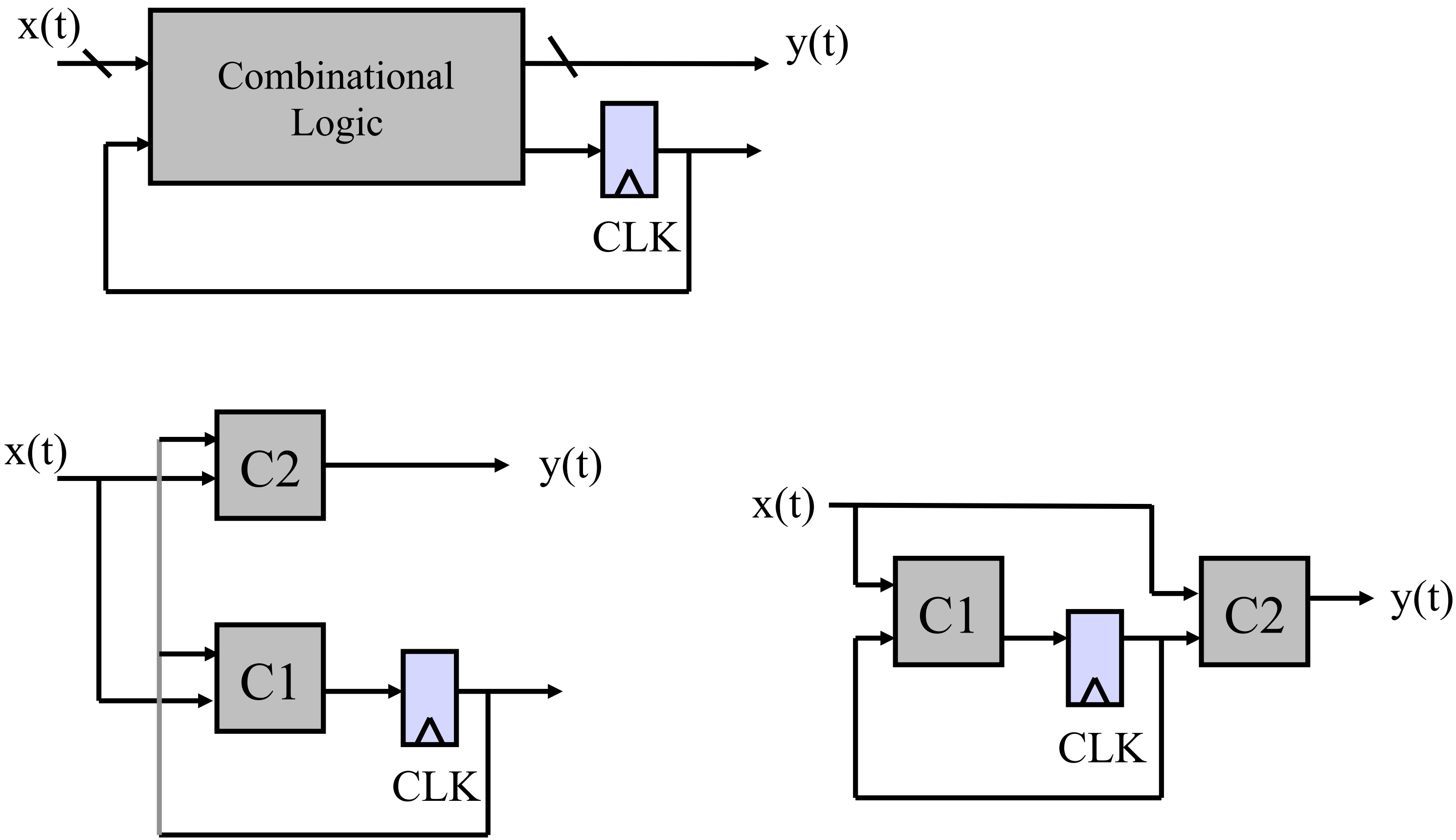
Time	0	1	2	3	4	5
Input	0	1	1	0	1	-
State	S0	S0	S2	S1	S3	S0
Output	0	0	0	0	1	0

Implementation

State Diagram => State Table => Logic Diagram

- Canonical Form: Mealy and Moore Machines
- Excitation Table
  - Truth Table of the F-F Inputs
  - Boolean algebra, K-maps for combinational logic
- Examples
- Timing

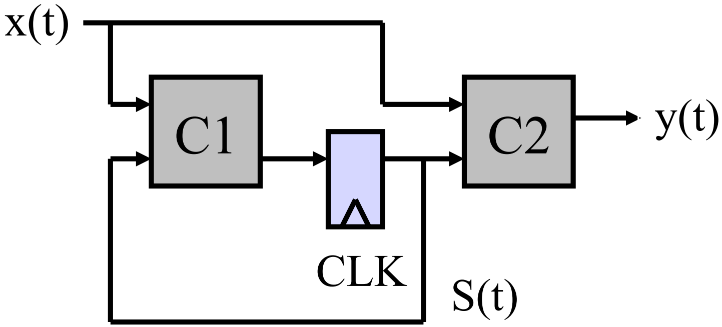
Canonical Form: Mealy and Moore Machines



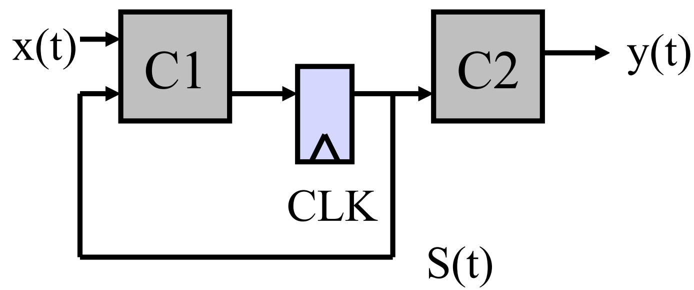
Canonical Form: Mealy and Moore Machines

Mealy Machine: $y_i(t) = f_i(X(t), S(t))$
Moore Machine: $y_i(t) = f_i(S(t))$

$s_i(t+1) = g_i(X(t), S(t))$



Mealy Machine



Moore Machine